10

Grid Computing for Artificial Intelligence

Yuya Dan
Matsuyama University
Japan

1. Introduction

This chapter is concerned in grid computing for artificial intelligence systems. In general, grid computing enables us to process a huge amount of data in any fields of discipline. Scientific computing, calculation of the accurate value of π, search for Mersenne prime numbers, analysis of protein molecular structure, weather dynamics, data analysis, business data mining, simulation are examples of application for grid computing. As is well known that supercomputers have very high performance in calculation, however, it is quite expensive to use them for a long time. On the other hand, grid computing using idle resources on the Internet may be running on the reasonable cost.

Shogi is a traditional game involving two players in Japan as similar as chess, which is more complicated than chess in rule of play. Figure 1 shows the set of *Shogi* and initial state of 40 pieces on 9 x 9 board. According to game theory, both chess and *Shogi* are two-person zero-sum game with perfect information. They are not only a popular game but also a target in the research of artificial intelligence.

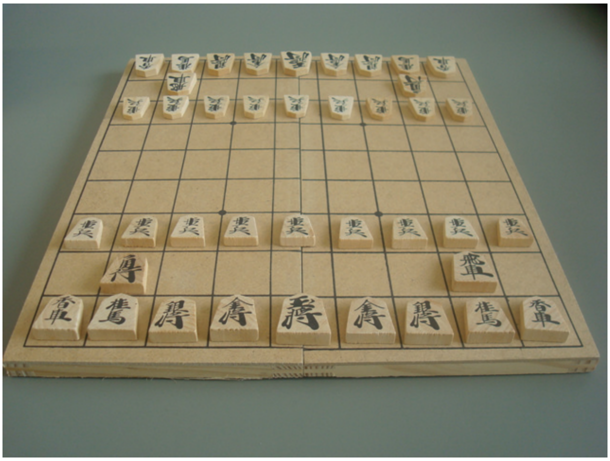

Fig. 1. Picture of *Shogi* set including 40 pieces on a 9 x 9 board. There are the corresponding king, rook, knight, pawn, and other pieces. Six kinds of pieces can change their role like pawns in chess when they reach the opposite territory.

The information systems for *Shogi* need to process astronomical combinations of possible positions to move. It is described by Dan (4) that the grid systems for *Shogi* is effective in reduction of computational complexity with collaboration from computational resources on the Internet. A solution to next move problems as artificial intelligence systems will be discussed in more detail in the chapter.

The chapter introduces the history and current state of computer *Shogi* systems, and the applicaion of grid computing for thinking routines for next move problems, then proposes an idea of the interaction between human and grid computing in game theory. Moreover, we suggest possibility of the human grid computing for artificial intelligence systems. It is described that the present situation of computer *Shogi* systems and the possibility of the human grid computing which assists the position evaluation function with human intuition.

Proposed model in the chapter seems to be applied to any other efficient systems. The wisdom of crowds can make up qualitatively for grid systems essentially based on the concept of Web 2.0. While the design of the human grid computing is only proposed, the development and the implementation of that are left open.

In the autumn of 2010, the grid system for *Shogi* is going to challenge against a human expert player of *Shogi*. The system is not human grid system, however, it is the first time to collect computing resources in order to play against a human expert player. We will also discuss the battle between human and computers in *Shogi*. Not only the human grid computing organized by amateur *Shogi* players but also the grid computing system constructed on the Internet may win against *Meijin* before 2015, when we predict as the X day.

2. Background

Grid computing is a great tool in order to solve problems in complex systems, because computing resources including Central Processing Unit (CPU), main memory, strage and so on are rich with a reasonable cost.

Grid technology is applied to large-scale computing, such as numerical simulation and data analysis not only in the academic research but also in the business world. Numerical simulation enables virtual experimentation on computers such as collision of rigid body, weather forecast in hydrodynamics, economic phenomenon, and so on. Data analysis is intended for huge amounts of data even for data mining with the purpose of marketing. SETI@home (22) is the most famous project for an application of data analysis by grid technology. At the present time, many computers are connected to the internet. The efficiency of machines has made remarkable progress more than we could possibly use their all resources. Idle resources of a large number of computers on the internet can realize high performance computing equivalent to supercomputers. We can say that we have already had an inexpensive supercomputer on the Internet.

We can use mathematics only for a simple model, however, cannot apply our hand-writing calculation to more complicated problems that have a large number of variables to be solved. In particular, combinatorial optimization has a difficult structure of problem that takes a very long time to finish searching the optimal solution. In other words, it is likely that we can solve the difficult problems with high performance computing beyond hand-writing calculation.

Natural sciences have been based the framework of mathematics to describe phenomena in the physical world. Newton formulated the law of motion with differential equations, and Einstein proposed the equation of gravitational fields with the concept of geometry. Almost all of the models of natural phenomena can be described for the corresponding ordinary or partial differential equations, so that we can analytically solve the equation with initial state and boundary conditions.

On the other hand, there are many unsolved models of differential equations even in the classical physics. Since a certain class of models is not solvable analytically, we calculate the alternative numeric solution with hand-writing approximation or computers. For example, the Navier-Stokes equation is a system of partial differential equations which describes the motion of a compressive and incompressive fluid in fluid mechanics. The equation is essentially nonlinear, and we can obtain no explicit solution to initial data. We can only calculate the numerical solution using finite element method etc.

Newton succeed in solving two body problems in gravitational fields, however, he were faced the difficulty in solving many body problems in the same framework. Differential equations know the motion of the Sun and the Earth interacting by gravitation, they cannot provide the clear solution with fundamental functions to the problem of the motion of the Sun, the Earth and the Moon exactly. We know the reason why mathematical analysis has these limitation in the existence of chaos which often appear in the nonlinear phenomena.

In the latter half of twentyth century, scientific comupation was made in a variety of fields. The contemporary sciences are based on theories, experiments and computational matter.

According to the progress of computers, we can use a large scale of computation for artificial intelligence based on the model of natural sciences. Althogh early reseaches of artificial intelligence failed to realize the ideas, supercomputers enables the models in artificial intelligence with high performance computing. Moreover, the number of computers on the Internet has grown like hotcakes, we obtain another choice for grid computing. There is possibility of high performance grid computing for artificial intelligence than cluster system in the present time.

In the following sections, we describe the history of research in artificial intelligence, construction of grid computing, and the application to the battle against human intelligence.

What are we able to do by grid technology? In other words, what does grid technology enable to do anything impossible so far? An answer proposed in the work of Dan (4) is to implement computer Shogi systems as artificial intelligence by human-computer interactions. That is another solution to the difficulty of astronomical combinations to be processed by only grid technology.

3. Artificial intelligence systems

3.1 History

Information scientists and engineers have struggled for realization of artificial intelligence, since Shannon (19) sugested the possibility of computer chess program with evaluating functions, which pointed out that a typical chess game involved about 10^{120} possible moves. It seems natural to let computers get hold of intelligence, and that we want to use computer resources instead of human labor. In fact, Shannon made the suggestion not only of computer chess, but also about design of electric circuit, translation from one language to another, strategic decisions in simplified military operations, and so on. Development of computer chess systems has been equipped with great academic interest to researchers in artificial intelligence.

John McCarthy and Marvin Minsky organized a summer workshop at Dartmouse College in 1956. There are ten researchers interested in the study of machine intelligence, artificial newral networks and automata theory. The Dartmouse workshop gave birth to a new science called artificial intelligence.

Year	Event
1950	Shannon proposed computer playing program
1956	Dartmouse workshop
1970	North American Computer Chess Championship
1986	Computer Shogi Association founded
1997	Deep Blue defeated Garry Kasparov
2002	3-Hirn System in Shogi
2009	Council System in Shogi
2010	Akara came out for a grid system

Table 1. History of Artificial Intelligence Research

The first working computer *Shogi* program was written in 1974 by a research group supervised by Takenobu Takizawa (20). The program did not cope well with the game-tree complexity, so it could only play at the level of beginners.

After that, the Computer Shogi Association (CSA) founded as a society for the study of computer *Shogi* programming in 1986. The CSA has motivated the leading researchers in the field and provided place to exchange ideas at academic symposiums such as the series of Game Programming Workshops.

Of course, the scope of research in artificial intelligence has not been limited to computer chess and *Shogi*. In that time, researchers have struggled the other problems in expert systems for medicine, artificial neural networks, pattern recognition, optimization or automation, classification, etc. However, it is not enough to run programs which need a plenty of resources on the computers in decades ago. Research in artificial intelligence have taken progress according to the progress of computing performance.

It follows from Moore's law that the densiy of semiconducters in electronic circuits grows two times for twelve to eighteen months. Since the throughput of computing is approximately proportional to the density of semiconducters, the program or software on artificial intelligence obtains higher performance in future, even if we do not anything else. Hence, we will have more effective performance when we brush up the algorhithm or discover new algorithm for solving difficuties in the research of high performance computing including artificial intelligence.

Table 1 shows the history of main events in the artificial intelligence researches.

3.2 Deep blue

Shannon's dream came true after a half of a century. The most remarkable epoch-making event for artificial intelligence occurred in May 1997 when the chess machine Deep Blue, developed by IBM research team, defeated world chess champion Garry Kasparov in an exhibition six-game match. This was a major success for computer game-playing and a milestone in the history of computer sciences.

Deep Blue is a parallel system with optimized algorithms designed for carrying out chess game tree searches. The system was composed of a 30-node IBMRS/6000 SP computer and 480 single-chip chess search engines, with 16 chess chips per SP processor. The SP system consists of 28 nodes with 120 MHz P2SC processors, and 2 nodes with 135 MHz P2SC processors. The nodes communicate with each other via a high speed switch. All nodes have 1 GB of Random Access Memory (RAM), and 4 GB of disk. The system ran the AIX 4.2 operating system.

The chess chips in Deep Blue are each capable of searching 2 to 2.5 million chess positions per second, and Deep Blue in total can examine and evaluate up to 200 million chess positions per second. The system can search a game tree to a depth of between six and twelve plies

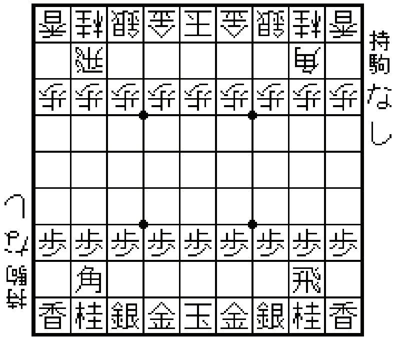

Fig. 2. Initial position on board in Shogi (Japanese chess)

to a maximum of forty plies in some situation, although the system speed varied widely, depending on the specific characteristics of the positions being searched. No one in the world can win against the strongest computer chess system even on ordinary PC at the present time. See Campbell et. al. (2), IBM web site (6), Hsu (9), Newborn (16) and (17) in detail for Deep Blue.

3.3 Difference between chess and *Shogi*

On the other hand, computer systems for *Shogi*, often referred to as Japanese chess, are behind those of chess in intelligence. *Shogi* is also a two-person zero-sum game with perfect information as chess. Iida, Sakuta and Rollason in (10) described the current state of the art for this game.

Let us consider the difference between chess and *Shogi*. First, Chess has an 8x8 board, while *Shogi* has an 9x9 board. There are 6 different pieces in chess, while there are 8 different kinds of pieces in *Shogi*. In chess each player has 32 pieces in total (16 pieces each), while in shogi each player has a total of 40 pieces (20 pieces each).

Second, in chess only the pawn is allowed to promote, otherwise in shogi promotion is allowed for 6 different kinds of pieces.

Third, difference between shogi and chess is the possibility of reusing pieces in shogi. When a piece is captured, the piece does not disappear from the game, but is put next to the board.

Finally, one of the crucial difference between chess and *Shogi* is the complexity of game-trees, which is significantly higher in *Shogi* than in chess. Captured pieces may be brought into play again by the player who captured the piece. Because *Shogi* has the ability to return captured pieces in hand from the opponent to the board, it generates a great number of combinations of each possible position in comparison with chess.

After all, *Shogi* has more spacious in game trees than chess does.

3.4 Game tree analysis

In order to solve next move problems, which returns the optimal move to given position, we analyze the game tree of position transition. The basic method is to find the solution for a player to win even if the opposite player will do the best. However, it is impossible to run full-width search to some sufficient depth. In fact, *Shogi* has 10^{219} possible search of game trees, while chess has 10^{123} in average. Table 2 shows comparison of complexity in Checker,

Game	Checker	Chess	*Shogi*	Go
Branching Factor	2.8	35	80	250
Average Game Length	70	80	115	150
Complexity	10^{31}	10^{123}	10^{219}	10^{360}

Table 2. Comparison of game-tree complexity in checker, chess, *Shogi*, Go

Chess, *Shogi* and *Go*, sorted by the average number of branching factor, average game length and game-tree complexity.

The game tree of *Shogi* is a graph of infinite number of vertices connected by directed egdes, which begin the root, that is, the vertix corresponds to the initial position. In order to solve next move problems, we should estimate the leaves where one player is check-mated. The game of *Shogi* finishes the state of check-mated, All most of ordinary routes of game tree from the root have end points.

For the average number of branching factor B and average game length L, the average search space that can be expected in a game is estimated by B^L. More precisely, the exact one is

$$G = \prod_{j=1}^{L} B(j) = B(1) \cdot B(2) \cdot \dots \cdot B(L) \tag{1}$$

where L denotes the average game length and $B(j)$ the number of branching factor of j-th ply for $j = 1, 2, \dots, L$. We know that $B(1) = 30$ and $B(2) = 30$ in *Shogi*, and $B(j)$ for $j \geq 3$ depends on the previous move and the present position. $B(j)$ becomes small when check-mated. It is known that *Shogi* has an average branching factor of about 80, and that the maximum branching factor is 593. It is also known that a game of *Shogi* takes about 115 plies on average, and that the maximum game length (in actual games) is more than 500 plies. See Matsubara et. al. (14) for *Shogi*, Allis (1) for the other games in detail.

Therefore, computer systems for *Shogi* have a long way to go before they will be able to offer a serious challenge to *Meijin*, a human champion of *Shogi*. In the newest research, it is expected that computer *Shogi* systems will defeat *Meijin* in 2015 at the earliest.

4. Computer *Shogi*

4.1 Algorithm for computer *Shogi*

The core algorithms for thinking routine of computer *Shogi* systems are analysis of the game tree and pruning using the alpha-beta method from min-max tree for evolutionary sequence of positions. These method to solve next move problems are similar to chess and other games. Given position information including array of pieces on the board, pieces in hands and the next player, the position evaluation function returns the value of the position forward several steps.

When initial position is given, all legal hands of the next player are genarated, say h_1 and h_2 for the hands. Then possible positions are P_1 and P_2 respectively. In order to select the best hand, all legal hands of the opposite player are genarated by recursive call. It is thought that players do their best, therefore the next hand will be selected by the min-max method under the restriction of finite moves.

Although a computer program seeks as many positions as possible, almost all of the evolutionary sequence seems not to be valid clearly even for amateur players. Full-width search algorithm can make sure a limited number of combination like checkers, resources of computer *Shogi* consume waste of time in many cases.

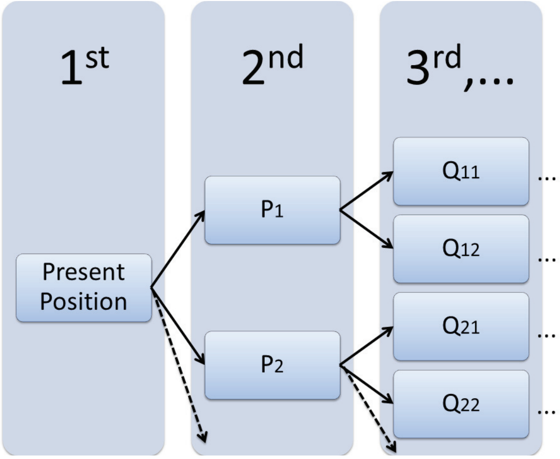

Fig. 3. Game trees present a evolutionary sequence of positions at playing games

There is a pseudo-code for the thinking routine of min-max method by recursive call in the appendix.

4.2 Position evaluation functions

We use position evaluation functions when we want to know or jugde which player is advantageous. Position evaluation functions return values from the status of position including array of pieces on the board, pieces in hands and the next player given as arguments. Position evaluation function are defined almost as static evaluation functions, constructed by heuristic process. These functions have a large number of parameters, so that these parameters are optimized automatically from the playing before. Machine learning is a method in artificial intelligence for the optimization.

The value for each pieces on the board is important factor to estimate the position. It may be different in time or position, we never know optimal values.

Let f be a real-valued (may take $\pm\infty$) function:

$$f : (p, h, t) \mapsto \mathbb{R} \cup \{\pm\infty\}, \tag{2}$$

where p denotes the array of position of all pieces on the board, h pieces in the hand of both players, t which is the next turn. $\pm\infty$ mean the status of check mated for each player, which is used in the program like this: For every evaluated real number v, $v < \infty$ and $-\infty < v$. The game ends up when the value of f takes $\pm\infty$, although a player decrare to finish the game before that in the real play.

In fact, position evaluation functions are also a function of "time". Opening game, middle game and end game are different each other in evaluation, computers feel difficulties in the evaluation of middle game. They are not good at overlooking on the board in the global point of view, however, almost perfect opening game for a huge scale of databases and end game for logical thinking.

4.3 *Shogi* using grid computing

Marsland and Popowich (13) proposed the parallel implementation of the alpha-beta search algorithm of game trees on distribution systems as a methematical model.

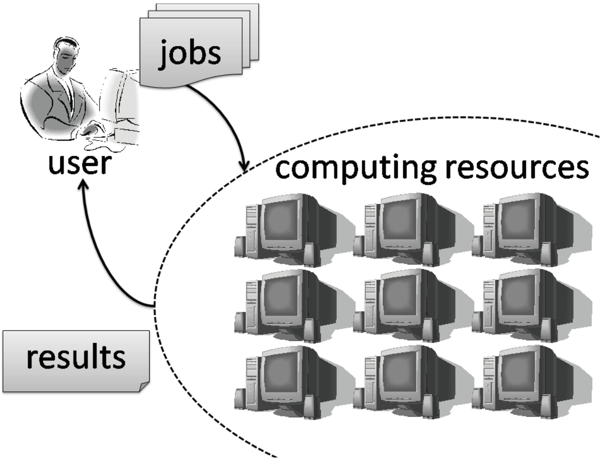

Fig. 4. A grid system is like a supercomputer but uses with computational resources on the Internet.

Grid computing is a possible technology solution for deeper search of the algorithms for computer *Shogi* systems. The system constructed by grid computing brings its forces into high throughput as super computers by a set of usual individual computers connected to the Internet. Although each computer does not have so high performance, the total performance of the connected computing resources surpasses super computers, when each individual computer is properly allocated to appropriate jobs.
If initial state of the *Shogi* position has B possible hands, one more deeper search will become possible by the B grid computers, two more deeper search by the B^2 grid computers. In general, the performance of grid systems improves at most logarithmically as N more deeper search needs B^N grid computers, regardless of bottleneck of job assignment.
On the other hand, $T = G/\rho$ where G in (1) is the number of node of game tree and ρ is the ability to calculate the position evaluation function per unit time, say $\rho = 200,000,000$ for Deep Blue.
Grid computing attains high throughput in enormous search of the optimized solution, however, it keeps bottleneck that the case of clearly misdirected search is not different from that without grid computing. Logarithmical improvement is hopeless for the full-width search in playing *Shogi*. Therefore, the other way to solve the problem is desired, which excludes clearly misdirected searches before evaluating the positions.
Figure 5 shows a next move problem in the Meijin title match held on 17 June, 2008. In the real play, Yoshiharu Habu use a pieces in hand near opposite King on the board which is not so easy, but there are many misdirected plies seen even if amateur Shogi players. So far computers cannot recognize the difference between promising and misdirected ply. They only can judge the amount of values of current position by calculating position evaluation functions.

4.4 Human grid computing

3-Hirn System (8) is to selecte the opinion of 2 advisors by one person. Advisors may be computers. This system is proposed for chess by Althofer in 1980, which is stronger than the person who is selecting one of the opinions proposed by two other advisors.

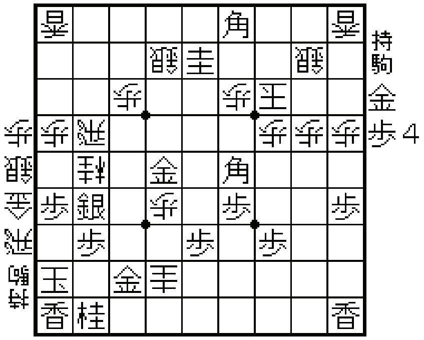

Fig. 5. The Next Move Problem

The 3-Hirn System is an example of interaction of human and computers, which is extended for a counsil system on playing game (7).

The computer Shogi by grid systems throws jobs to client computers on the internet. Such client computers run the position evaluation functions, as they display the process and the result of calculation, and then return the value of the evolutionary sequence of positions back to the grid server. It improves efficiency in enormous search of the next move, if amateur Shogi players in front of the screens dismiss negative results.

When an amateur Shogi player j can dismiss negative results per unit time, and we denote by ρ_j. For the average time T to think one ply, total possible positions are reduced to

$$G - \sum_{j=1}^{M} T\rho_j \tag{3}$$

where M is the number of amateur Shogi players, regardless of the correctness. This formula means we can reduce total possible positions from G, as M is large and ρ_j is small.

Therefore, we conclude the relation

$$T\left(\rho + \sum_{j=1}^{M} \rho_j\right) = G. \tag{4}$$

It should be remarked that $\rho_j << \rho$ for $j = 1, 2, \cdots, M$.

On the other hand, evaluation system for human intuition is indispensable. All of human intuition are not always wise as we know. It is necessary for human grid systems that the judge of amateur Shogi players is evaluated by achievements. The position evaluation functions include the information who judge the negative results as arguments.

Instead of human assisted system, the proposed system has possibility of realizing high-performance intelligence ever.

5. Implementation environment

Although there are lots of computing resources on the Internet, we do not use that without the permission of their owners. The performance of grid systems depends on the number of computational resources on the Internet.

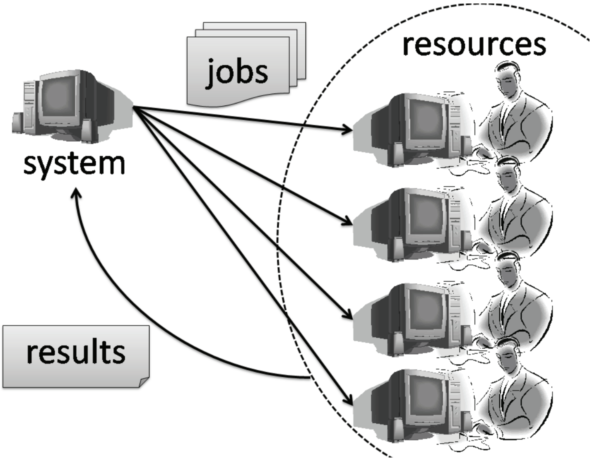

Fig. 6. Human Grid

Let us consider the number of computing resources constructed for grid systems, following from the discusstion on the growth of social networks according to Dan (5). We will use a mathematical model of diffusion on social networks.
Let N be the total number of the computing resources on the Internet to be estimated which is a positive constant for a while in our discussion, $n(t)$ be the number of active computing resources for use which is also a function of time t. The inequality

$$0 \leq n(t) \leq N \tag{5}$$

holds for any t in $\mathbb{R}$. According to the property of growth and the assumption that active ones does not run away from the grid system constructed, it can be supposed that $n(t)$ increases as t goes.
Continuation can extend $n(t)$ be a continuous function of t from the domain of integers. According to the process, the probability of the growth is proportional to both $n(t)$ and $N - n(t)$. We can estimate the increase rate of active people as follows:

$$\frac{dn(t)}{dt} = \lambda n(t)\,(N - n(t)) \tag{6}$$

with a positive constant λ. In the right hand side of this equation, teh terms $n(t)$ and $N - n(t)$ denotes incoming and outgoing effect respectively. The differential equation (6) is derived from our model for the growth of the number of computing resources on the Internet.
Then, we obtain from the form of separation of variables,

$$\frac{1}{N}\left(\frac{1}{n(t)} - \frac{1}{n(t)-N}\right)\, dn = \lambda\, dt. \tag{7}$$

Integrating both side of the equation, we have

$$\frac{1}{N}\log\left|\frac{n(t)}{n(t)-N}\right| = \lambda t + C \tag{8}$$

with a integral constant C. This means

$$\frac{n(t)}{N - n(t)} = e^{\lambda N(t-t_0)}, \tag{9}$$

where

$$t_0 = -\frac{C}{\lambda N} \tag{10}$$

which is determined by initial conditions. Finally, we can solve $n(t)$ as

$$n(t) = \frac{N}{1 + e^{-\lambda N(t-t_0)}}, \tag{11}$$

which is known as one of the logistic curves. It should be remarked that $n(t)$ is equal to $N/2$ when $t = t_0$ and $n(t)$ is equal to N when $t \to \infty$. From the differential equation (6), $n(t)$ increases the most rapidly when $t = t_0$.

If the total number of the Internet is not constant, say does increase, then we should consider another differential equation for the spread of grid computing. We can extend (refDifferentialEquation) to more general expression:

$$\frac{dn(t)}{dt} = \lambda n(t)\left(N(t) - n(t)\right), \tag{12}$$

where N(t) is not a constant, but the function of t. We cannot solve the equation above analytically, all we have to do is the numeric calculation using computers to obtain the approximate solutions.

Because of the comparison of growth rates, that is,

$$\frac{dn(t)}{dt} = \lambda n(t)\left(N(t) - n(t)\right) \geq \lambda n(t)\left(N - n(t)\right), \tag{13}$$

it is easy to prove that the solution to the equation (12)

$$n(t) \geq \frac{N}{1 + e^{-\lambda N(t-t_0)}}, \tag{14}$$

if $N(t) \geq N$ for $t \in \mathbb{R}$. We can expect that the population $n(t)$ is greater or equal to the above eatimation.

6. Akara 2010

It is a difficult question what a computer is even for the human experts in computer science and philosophy.

Akara 2010 is a grid or cluster system for Shogi as the project of 50th anniversary of Information Processing Society in Japan (IPSJ). It is constructed for a council system of four different Shogi programs on 169 Intel Xeon CPUs with 676 cores in the University of Tokyo as a cluster system. The computers are usually used for the class of exercises for the students at the university, which is not constructed for the particular purposes in high performance computing.

The name of Akara stands for 10^{224} in Japanese, which is the around the number in average of possible moves for one game of Shogi. We, researchers in artificial intelligence, were expected that Akara might be the first information system for winning against the professional human Shogi player.

On the other hand, Ichiyo Shimizu is the top of the woman Shogi players who takes the match against Akara 2010. She is not the top or the champion in Shogi, however, if Akara 2010 win against her, then Akara series will be permited to challenge the strongest Shogi player. The goal of the project is the day when Akara defeats the human Shogi champion as Deep Blue did in 1997. Since Akara won against Shimizu, we can see the challenge with Akara 2011 in the next chance.

On 11th October in 2010, Akara 2010 defeated Shimizu in 86 plies. Three hours to think are given to both of them.

7. Concluding remarks

The chapter suggests the progress of the grid computing and the possibility of human grid computing for artificial intelligence systems for Shogi. It is described that the present situation of computer Shogi systems and the possibility of the human grid computing which assists the position evaluation function using human intuition. In the most advanced systems, position evaluation functions are constructed by the parameters which are determined by the technology of machine learning. We can see that the next move problems are the optimal combinatorial ones under the rules of Shogi.

According to the history of computer chess, advances not only in hardware but also in software algorithm enable the breakthrough for artificial intelligence. It is the time that we expect to consider the challenge from the computer in formal. Deep Blue had an extreme performance in the search of game trees in chess specially for the human champion, so that the full-width search was very efficient at the challenge to Kasparov.

We also introduced Akara 2010 in the project of game computing in Japan, which is expected as the first computer program to win against the human champion of Shogi. Computing performance have grown up in the exponential rate of Moore's law, so that it is certain to exceed the strength of human abilities, although the information systems for Shogi need to process astronomical number of combinations of possible positions to move.

Cluster computing is gathering the computing resources efficiently, where we can expect the linear growth of performance proportional to the number of computers in cluster system. Grid computing has difficulties in the growth of performance efficiently, however, we expect to gather a large number of computing resources in the Internet. Grid computing faces at the limitation we estimated in the theory of growth, or logistic curve generalized.

On the other hand, human grid computing cannot be a pure artificial intelligence system. However we expect the reduction of the game tree search or investigation in Shogi. Human intuition or global viewpoint is blind at the clearly misdirected search.

The wisdom of crowds can make up qualitatively for grid systems essentially based on the concept of Web 2.0. While the design of the human grid computing is proposed, the development of that remains open.

The human grid computing organized by amateur Shogi players may win against Meijin before 2015. We can also expect that the grid or cluster system Akara defeats series the Meijin in recent future.

8. Acknowledgement

The author would like to thank Dr. Peter Dell, Dr. Hitoshi Okada and Prof. Shiro Uesugi for useful suggestions and improvements on the earlier version of the work at the ITeS workshop of the 2008 International Symposium on Applications and the Internet (SAINT 2008) in Turku, Finland. He also thanks Takehiro Moriya for fruitful discussion on the most advanced grid computing technology.

9. References

[1] L. V. Allis, *Searching for Solutions in Games and Artificial Intelligence*, Ph.D. thesis, University of Limburg, Maastricht. (1994)

[2] M. Campbell, A. J. Hoane Jr., F. Hsu, "Deep Blue," *Artificial Intelligence*, Vol. 134, Iss. 1–2, pp. 57–83. (2002)

[3] Condor, http://www.cs.wisc.edu/condor/

[4] Y. Dan, "Reduction of misdirected search in game trees by human grid computing," Journal of Informatics and Regional Studies, pp. 35–42. (2009)

[5] Y. Dan, "Modeling and simulation of diffusion phenomena on social networks," to apear in *Proceedings of 2011 Third International Conference on Computer Modeling and Simulation (ICCMS 2011)*.

[6] Deep Blue (IBM), http://researchweb.watson.ibm.com/deepblue/

[7] M. Hanawa, T. Ito, (in Japanese) "The Council System on Playing Game," *The third simposium of entertainment and cognitive science*. (2009)

[8] T. Hashimoto, M. Sakuta, H. Iida, "3-Hirn System: The First Results in Shogi," *Game Programming Workshop in Japan*, pp. 57–64. (2002)

[9] F. Hsu, *Behind Deep Blue*, Princeton University Press, New Jersey. (2002)

[10] H. Iida, M. Sakuta, J. Rollason, "Computer Shogi," *Artificial Intelligence*, Vol. 134, Iss. 1–2, pp. 121–144. (2002)

[11] D. E. Knuth, R. W. Moore, "An Analysis of Alpha-Beta Pruning," *Artificial Intelligence*, Vol. 6, No. 4, pp. 293–326. (1975)

[12] D. N. L. Levy, M. Newborn, *All about Chess and Computers*, Computer Science Press. (1982)

[13] T .A. Marsland, F. Popowich, "Parallel game-tree search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 4, pp. 442–452. (1985)

[14] H. Matsubara, K. Handa, (in Japanese) "Some Properties of Shogi as a Game," *Proceedings of Artificial Intelligence*, Information Processing Society of Japan, Vol. 96, No. 3, pp. 21–30. (1994)

[15] H. Matsubara, H. Iida, R. Grimbergen, "Chess, Shogi, Go, natural developments in game research," *ICCA Journal*, Vol. 19, No. 2, pp. 103–112. (1996)

[16] M. Newborn, *Kasparov Versus Deep Blue: Computer Chess Comes of Age*, Springer-Verlag, New York. (1997)

[17] M. Newborn, *Deep Blue: an artificial intelligence milestone*, Springer-Verlag, New York. (2003)

[18] T. O'Reilly, "What Is Web 2.0," http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[19] C. E. Shannon, "Programming a computer for playing chess," *Philosophical Magazine*, Vol. 41, pp. 256–275. (1950)

[20] T. Takizawa, (in Japanese) "Report on Computer-Shogi Grand Prix in 1999," *Journal of Computer Shogi Association*, Vol. 12, pp. 39–41. (1999)

[21] H. Yamashita, "Algorithm of move generation in a shogi program," *Proceedings of The Game Programming Workshop*, Computer Shogi Association, pp. 134–138. (1994)

[22] SETI@home, http://setiathome.ssl.berkeley.edu/

[23] IPSJ project of computer Shogi, http://www.ipsj.or.jp/50anv/shogi/index2.html
[24] Komazakura, http://komazakura.shogi.or.jp/vscomshogi/kifu/

10. appendix

```
int thought( int Level, int *Board, int NextPlayer, int *BestMove )
{
  int i, j;
  int n;
  int CriticalValue, value;
  int BetterMove, MoveList[594];
  int CheckMate = isCheckMated( Board, NextPlayer );
  int x, y;
  int koma;

  if( CheckMate != 0 )
    return CheckMate * ( Level + 1 ) * 32;

  n = GenerateMove( Board, NextPlayer, MoveList );
  if( n == 0 )
    return EvaluatePosition( Board );

  if( Level == 0 )
    return n;

  BetterMove = MoveList[0];
  CriticalValue = -1<<16;
  for( i = 0; i < n; i++ ){
    koma = MoveTo( Board, MoveList[i], k );
    value = k * thought( Level - 1, Board, NEXT( k ), BestMove );
    if( value > CriticalValue ){
      CriticalValue = value;
      BetterMove = MoveList[i];
    }
    Board[MoveList[i] / 256] = Board[MoveList[i] % 256];
    Board[MoveList[i] % 256] = koma;
  }

  *BestMove = BetterMove;
  return ( ( k == SELF ) ? 1 : -1 ) * n;
}
```